Lesson Plan

Lesson 2: Beating heart

|  |  |
| --- | --- |
| **Ages:** 7 – 11 |  |
| **Programming language:** MakeCode blocks |
| **Topics**: Algorithms (Computational thinking), Iteration/loops (Programming)  **Outcomes:** Students start to learn about sequences and loops by making simple animations on the micro:bit’s LED display. |

### Key learning in this lesson

* Understand that sequence and timing is important when making an animation.
* Understand that animations create an illusion of movement by showing a sequence of still images.
* Code the micro:bit to show simple animations on its LED display output.
* Use loops to make animations run longer using fewer instructions.

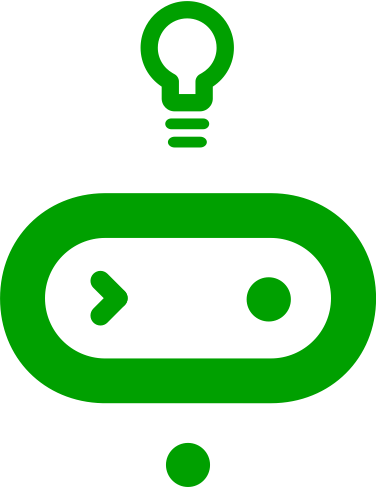
### Learning objectives

* I can create a micro:bit animation using a sequence of images in a loop.
* I can explain that the order or sequence of instructions is important.
* I can explain that loops can make code more compact and easier to read.

# Preparation: before the lesson

### What you need

* BBC micro:bits and micro-USB cables – at least one for every two students
* At least one computer (laptop or desktop) for every two students, with internet access to the Microsoft MakeCode editor: <https://makecode.microbit.org/>
* Alternatively, you can use iPads with the micro:bit app installed. See our guide: <https://mbit.io/lessons-mobile>
* micro:bit battery packs (optional) – one per micro:bit
* PowerPoint presentation – whole class teaching slides
* Code blocks student handout (optional)
* LED planning sheet (optional for extension task)

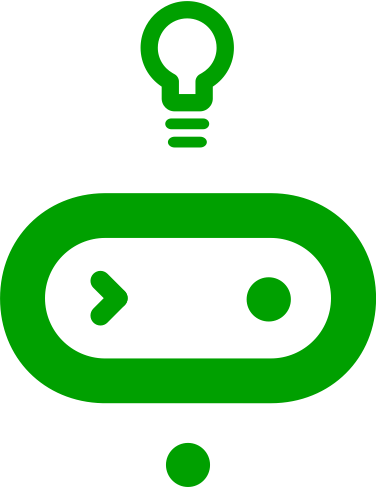
****The lesson download also includes an optional ‘.hex’ program file of the completed project, which may be useful if you have limited internet access. You can drag and drop this direct onto the MICROBIT drive when you connect a micro:bit to your computer. You can also drag ‘.hex’ files into the MakeCode editor to examine the code and test it in the simulator.

### Differentiation ideas

* If this is one of your first coding lessons with the micro:bit, it may be hard to know which students will need more support. You can use the extension ideas in the teaching section below for students who complete the task more quickly than others.

### Decide how to deliver the ‘Create’ coding activity

You’ll share the completed code on screen with your whole class from the slides. You can choose any of these methods that suit your classroom and teaching style:

* You (or selected students) model building and testing the code yourself on a large screen. The completed code is in the lesson plan and slide deck.
* Give students printed code blocks handouts to follow or cut out and assemble.
* Share a step-by-step YouTube coding video with the whole class, or individual students.
* If YouTube is blocked in your school, we also provide an animation in the slides showing how to assemble the code.
* Students can individually follow an online step-by-step tutorial.
* ****You can also choose to manage the whole class coding activity and save every student’s code using **micro:bit classroom**. Find out more at <https://mbit.io/lessons-classroom>

### Decide how to deliver the ‘Evaluate’ activity

Students download their code to real micro:bits and test the project.

You may want your students to answer the evaluation questions:

* on paper
* verbally with partners
* as part of a whole-class discussion.

### Glossary

|  |  |
| --- | --- |
| **algorithm**: | a set of step-by-step instructions |
| **animation**: | creating the illusion of movement by showing a sequence of still images |
| **icon**: | a graphic representation of something on a computer screen |
| **infinite loop**: | a loop that runs forever |
| **LED**: | light emitting diode - the micro:bit display is made of 25 LEDs |
| **loops**: | allow you to repeat sets of instructions without having to write them out multiple times |
| **output**: | data sent from a computer such as words shown on the display |
| **sequence**: | the specific order of instructions |

# Teaching: during the lesson

### Image of slide 2. Recap prior learning (slide 2)

Ask your students what they discovered about the micro:bit last time, for example:

* it’s a tiny computer.
* you tell it what to do by creating code (also called programs or software) and sending your code to the micro:bit.
* the micro:bit has an LED display which it uses as an output, sending out information, like our names, we have chosen, that people can read and see.
* the ‘forever’ loop kept the code running.

## Think: starter activity

### Image of slide 3.Learning objectives (slide 3)

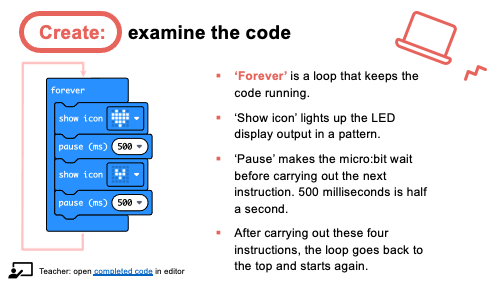
* Share thelearning objective,‘I can create a micro:bit animation using a sequence of images in a loop.’
* Explain: animation is a sequence of still images shown one after the other to look like movement.
* Loops allow us to repeat sets of instructions without having to write them out multiple times.
* Ask students: where else you have seen animations? E.g., cartoons, games.

### Beating heart introduction video (slide 4)

* Optionally play project introduction video: <https://mbit.io/lessons-heart-intro-video>

## Create: coding activity

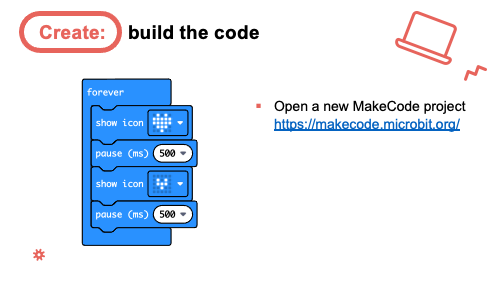
### Examine code with students (slide 5)

Explain:

* The ‘forever’ block is a **loop** which keeps the sequence going. It’s an ‘infinite loop’ – a loop that keeps going as long as the micro:bit has power.
* The ‘show icon’ and ‘pause’ blocks make up the **sequence**. Changing the pause time makes the animation faster or slower.
* At the end of the sequence, the loop goes back to the top and starts again. Using a loop means we can use only 4 blocks to create an animation that runs forever.

You can optionally follow the link in slide 5 to open the completed code in the editor: <https://mbit.io/lessons-heart-code>   
You can model changing the pause times and seeing what effect that has in the simulator.

### Model building the code (slides 6-9)

* You can open a new MakeCode project from slide 6 and model building the code from scratch. All the blocks you need are in the ‘Basic’ section.
* Optionally share the YouTube coding video on slide 7 with the class: <https://mbit.io/lessons-heart-code-video>
* Or share the coding animation on slide 8 if YouTube is blocked in your school.

Students recreate the code, testing it in the simulator. They can:

* Copy the code from slide 6.
* Follow printed code blocks handouts.
* Individually follow a step-by-step online tutorial:  
  <https://mbit.io/tutorial-beating-heart> - you can share the link from slide 9
* If you are using [micro:bit classroom](https://classroom.microbit.org/), start a new session and ask your students to join your lesson. You can also open a session with completed code to edit and share with students: <https://mbit.io/lessons-heart-classroom>

## Screen shot of slide 10Evaluate: (slide 10)

Students transfer code to their micro:bit and test.

**Questions:**

* Does it work as you expect?
* If not, do you need to debug the code and download it again?
* How good is the project?
* Would you recommend it to a friend?
* How could you improve it?
* Could it have other uses?
* How does it work?
* Encourage students to think about how it works when holding it in their hands.

## Image of slide 11. Extend: (optional, slide 11)

If students finish early, they can remix their code:

* Change the pause times – what effect does shorter and longer pause times have on the animation? Shorter pauses (smaller numbers) make the animation faster. Longer pauses (bigger numbers) make the animation slower.
* Create your own animations using the ‘show LEDs’ block which lets you draw your own pictures. You can use the LED planning sheet download to plan animations on paper before using the computer.
* Make longer sequences with more images.

## Share: revisit learning objectives (slide 12)

* I can create a micro:bit animation using a sequence of images in a loop.
* I can explain that the order or sequence of instructions is important.
* I can explain that loops can make code more compact and easier to read.

 **Ask**:

* Did your **sequence** of pictures and pauses make an image appear to move? (Pauses are part of the sequence. If they’re in the wrong place, or are too short, you won’t see the images. If pauses are too long, the illusion of movement is broken. Also, if you have more than two images, the sequence they are shown in is very important.)
* How does a loop help? (The loop keeps the animation sequence running forever, but without adding any extra code blocks.)

## Image of slide 13. Next steps (slide 13)

Today we used the ‘forever’ **loop** block and **sequences** of instructions to make **animations** on the micro:bit’s LED display output.

Next time we’ll use some of the micro:bit’s **inputs**, to make different pictures appear when we press different buttons.

# Assessment: after the lesson

When assessing students’ work you may find it helpful to ask these questions:

|  |  |
| --- | --- |
|  | Did the student make an animation work in practice using a sequence of images and a loop, and transfer it to their micro:bit? |
|  | What is their understanding of the importance of sequence, including pauses, in making a successful animation? |
|  | Can they explain the benefits of using a loop in the animation code? |

**Here are some guiding criteria you might want to include when assessing your students’ work:**

### WORKING TOWARDS the learning objective

|  |  |
| --- | --- |
|  | The student created some code, either in the simulator or on a real micro:bit, that showed at least one image on the display. |
|  | They can explain that different images are needed but not that the order and timing of an animation sequence is important. |
|  | They can explain that a loop keeps the animation running, but cannot explain other benefits of using loops, for example using fewer blocks and making the code simpler and more compact. |

### MEETING the learning objective

|  |  |
| --- | --- |
|  | The student created an animation in code using built-in icon images a loop and transferred it to their micro:bit. |
|  | The student can explain that the order and timing of an animation sequence is important to create the illusion of movement. |
|  | They can explain that loops keep an animation running without adding extra code to show more images, and loops make the code easier to read. |

### EXCEEDING the learning objective

|  |  |
| --- | --- |
|  | The student created animation sequences, which they transferred to their micro:bit. They also added other built-in images or their own image designs, or they created longer and more complex animation sequences in a loop. |
|  | They can explain that the order and timing of an animation sequence is important to create the illusion of movement, for example that if the images are shown too quickly, they can’t be seen, or if the sequence is too slow, the illusion of movement is broken. |
|  | They can explain that loops keep an animation running without adding extra code to show more images, and that loops make programs easier to read, understand and modify. |